

# Class 8: Breast Cancer Mini Project

Zoe Matsunaga (PID: A16853288)

## Table of contents

Background . . . . .	1
Data Import . . . . .	1
Clustering . . . . .	4
Principal Component Analysis . . . . .	6
Importance of Data Scaling . . . . .	6
PCA wisc.data . . . . .	10
Combining Methods . . . . .	15
Clustering on PCA Results . . . . .	16
Prediction . . . . .	17

## Background

This class mini-project focuses on unsupervised learning analysis of human breast cancer cell data. Principal component analysis (PCA) for dimensionality reduction will be conducted and then hierarchical and k-means clustering techniques will be applied. The project involves exploratory data analysis, interpreting PCA results, evaluating clustering performance by comparing cluster assignments to actual diagnoses, and optionally combining PCA with clustering. The goal is to identify potential groupings within the cell data based on their characteristics without prior knowledge of malignancy, and the project concludes with an application of the PCA model to classify new patient samples.

## Data Import

Our data from the U. of Wisconsin Medical Center

```
wisc.df <- read.csv("WisconsinCancer.csv", row.names=1)
head(wisc.df)
```

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean
842302	M	17.99	10.38	122.80	1001.0
842517	M	20.57	17.77	132.90	1326.0
84300903	M	19.69	21.25	130.00	1203.0
84348301	M	11.42	20.38	77.58	386.1
84358402	M	20.29	14.34	135.10	1297.0
843786	M	12.45	15.70	82.57	477.1
	smoothness_mean	compactness_mean	concavity_mean	concave.points_mean	
842302	0.11840	0.27760	0.3001		0.14710
842517	0.08474	0.07864	0.0869		0.07017
84300903	0.10960	0.15990	0.1974		0.12790
84348301	0.14250	0.28390	0.2414		0.10520
84358402	0.10030	0.13280	0.1980		0.10430
843786	0.12780	0.17000	0.1578		0.08089
	symmetry_mean	fractal_dimension_mean	radius_se	texture_se	perimeter_se
842302	0.2419		0.07871	1.0950	0.9053
842517	0.1812		0.05667	0.5435	0.7339
84300903	0.2069		0.05999	0.7456	0.7869
84348301	0.2597		0.09744	0.4956	1.1560
84358402	0.1809		0.05883	0.7572	0.7813
843786	0.2087		0.07613	0.3345	0.8902
	area_se	smoothness_se	compactness_se	concavity_se	concave.points_se
842302	153.40	0.006399	0.04904	0.05373	0.01587
842517	74.08	0.005225	0.01308	0.01860	0.01340
84300903	94.03	0.006150	0.04006	0.03832	0.02058
84348301	27.23	0.009110	0.07458	0.05661	0.01867
84358402	94.44	0.011490	0.02461	0.05688	0.01885
843786	27.19	0.007510	0.03345	0.03672	0.01137
	symmetry_se	fractal_dimension_se	radius_worst	texture_worst	
842302	0.03003		0.006193	25.38	17.33
842517	0.01389		0.003532	24.99	23.41
84300903	0.02250		0.004571	23.57	25.53
84348301	0.05963		0.009208	14.91	26.50
84358402	0.01756		0.005115	22.54	16.67
843786	0.02165		0.005082	15.47	23.75
	perimeter_worst	area_worst	smoothness_worst	compactness_worst	
842302	184.60	2019.0	0.1622		0.6656
842517	158.80	1956.0	0.1238		0.1866
84300903	152.50	1709.0	0.1444		0.4245
84348301	98.87	567.7	0.2098		0.8663
84358402	152.20	1575.0	0.1374		0.2050
843786	103.40	741.6	0.1791		0.5249
	concavity_worst	concave.points_worst	symmetry_worst		

842302	0.7119	0.2654	0.4601
842517	0.2416	0.1860	0.2750
84300903	0.4504	0.2430	0.3613
84348301	0.6869	0.2575	0.6638
84358402	0.4000	0.1625	0.2364
843786	0.5355	0.1741	0.3985
fractal_dimension_worst			
842302	0.11890		
842517	0.08902		
84300903	0.08758		
84348301	0.17300		
84358402	0.07678		
843786	0.12440		

Q1. How many patients/smamples are in this dataset?

```
nrow(wisc.df)
```

```
[1] 569
```

Q2. How many of the observations have a malignant diagnosis?

```
table(wisc.df$diagnosis)
```

```

  B   M
357 212

```

```
sum(wisc.df$diagnosis == "M")
```

```
[1] 212
```

Q3. How many variables/features are in the data are suffixed with \_mean?

```
grep("_mean", colnames(wisc.df), value=T)
```

```

[1] "radius_mean"      "texture_mean"     "perimeter_mean"
[4] "area_mean"        "smoothness_mean"  "compactness_mean"
[7] "concavity_mean"   "concave.points_mean" "symmetry_mean"
[10] "fractal_dimension_mean"

```

```
length(grep("_mean", colnames(wisc.df), value=T))
```

```
[1] 10
```

There is a diagnosis column that is the clinician consensus that I want to exclude from further analysis. We will come back later and compare our results to this diagnosis.

```
diagnosis <- as.factor(wisc.df$diagnosis)
head(diagnosis)
```

```
[1] M M M M M M
Levels: B M
```

Now we can remove it from the `wisc.df`

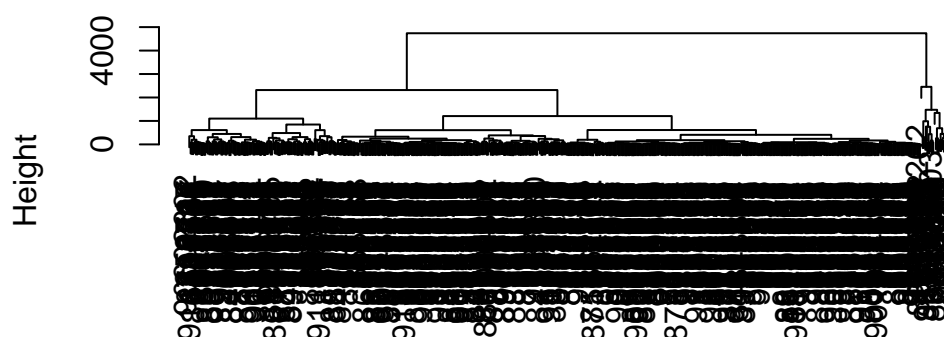
```
wisc.data <- wisc.df[,-1]
```

## Clustering

Let's try `hclust()`

```
hc <- hclust(dist(wisc.data))
plot(hc)
```

## Cluster Dendrogram



```
dist(wisc.data)
hclust (*, "complete")
```

We can extract clusters from this rather poor dendrogram/tree with the `cutree()`

```
grps <- cutree(hc, k=2)
```

How many individuals in each cluster?

```
table(grps)
```

```
grps
 1  2
549 20
```

```
table(diagnosis)
```

```
diagnosis
 B  M
357 212
```

We can generate a cross-table that compares our cluster `grps` vector without `diagnosis` vector values

```
table(diagnosis, grps)
```

```
      grps
diagnosis 1  2
B 357    0
M 192    20
```

## Principal Component Analysis

### Importance of Data Scaling

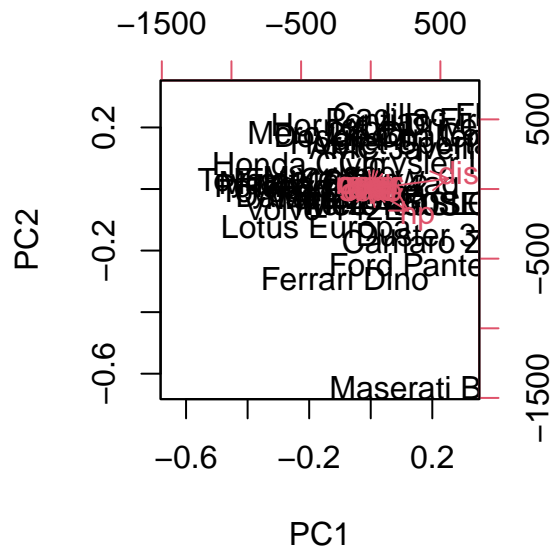
The main function for PCA in “base” R is `prcomp()`. It has a default input parameter of `scale=FALSE`.

```
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

We could do a PCA of this data as is and it could be misleading...

```
pc <- prcomp(mtcars)
biplot(pc)
```



Let's look at the mean values of each column and their standard deviation.

```
colMeans(mtcars)
```

mpg	cyl	disp	hp	drat	wt	qsec
20.090625	6.187500	230.721875	146.687500	3.596563	3.217250	17.848750
vs	am	gear	carb			
0.437500	0.406250	3.687500	2.812500			

```
apply(mtcars, 2, sd)
```

mpg	cyl	disp	hp	drat	wt
6.0269481	1.7859216	123.9386938	68.5628685	0.5346787	0.9784574
qsec	vs	am	gear	carb	
1.7869432	0.5040161	0.4989909	0.7378041	1.6152000	

We can “scale” this data before PCA to get a much better representation and analysis of all the columns.

```
mtscale <- scale(mtcars)
```

```
round(colMeans(mtscale))
```

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	0	0	0	0	0	0	0	0	0	0

```
apply(mtscale, 2, sd)
```

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	1	1	1	1	1	1	1	1	1	1

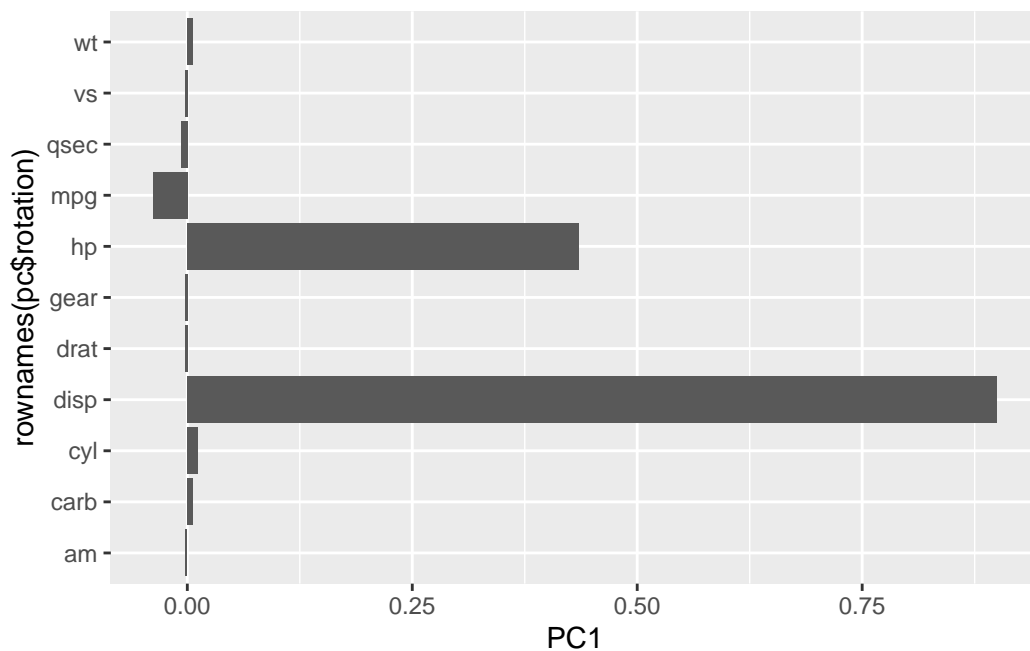
```
pc.scale <- prcomp(mtscale)
```

We can look at the two main results figures from PCA - the “PC plot” (a.k.a score plot, ordination plot, or PC1 vs PC2 plot, etc). The “loadings plot” how the original variables contribute to the new PCs.

A loading plot of the unscaled PCA results:

```
library(ggplot2)
```

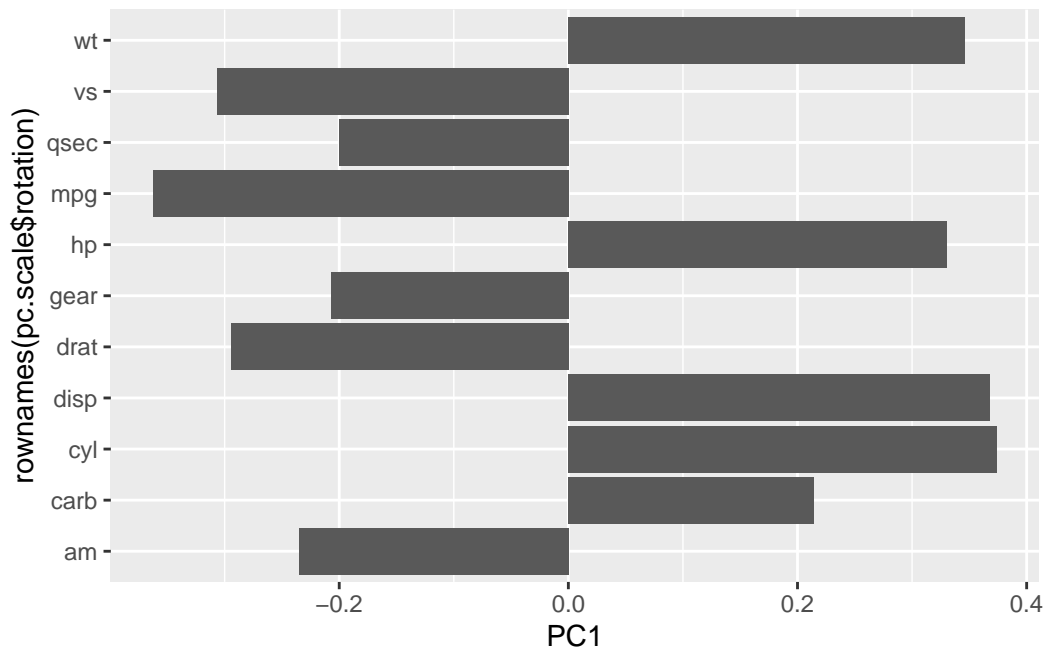
```
ggplot(pc$rotation) +  
  aes(PC1, rownames(pc$rotation)) +  
  geom_col()
```





Loading plot of the scaled data.

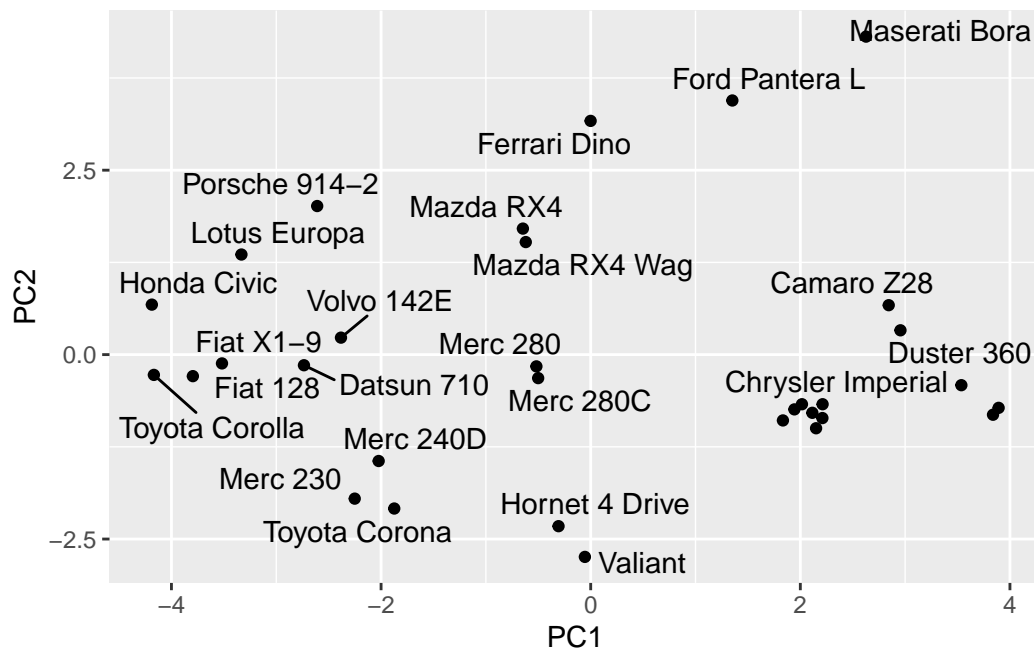
```
ggplot(pc.scale$rotation) +  
  aes(PC1, rownames(pc.scale$rotation)) +  
  geom_col()
```



PC plot of scaled PCA results:

```
library(ggrepel)  
ggplot(pc.scale$x) +  
  aes(PC1, PC2, label=rownames(pc.scale$x)) +  
  geom_point() +  
  geom_text_repel()
```

Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider increasing max.overlaps



**Key Point:** In general we will set `scale=TRUE` when we do PCA. This is not the default but probably should be...

We can check the SD and mean of the different columns in `wisc.data` to see if we need to scale - hint we do!

### PCA wisc.data

```
wisc.pr <- prcomp(wisc.data, scale=TRUE)
```

To see how well PCA is doing here in terms of capturing the variance/spread in the data we can use the `summary()` function.

```
summary(wisc.pr)
```

Importance of components:

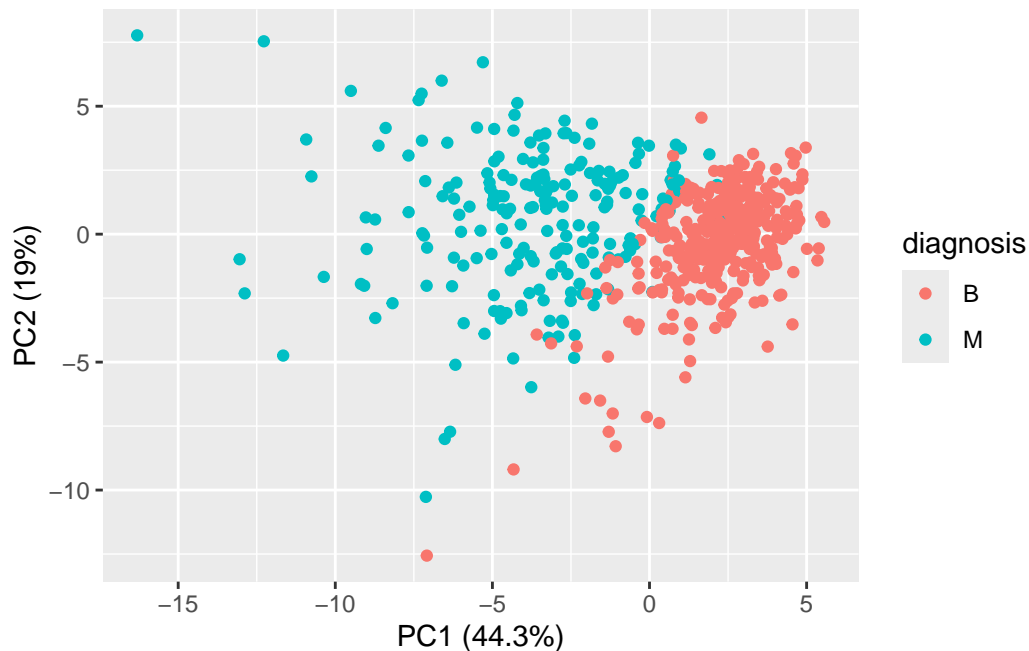
	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	3.6444	2.3857	1.67867	1.40735	1.28403	1.09880	0.82172
Proportion of Variance	0.4427	0.1897	0.09393	0.06602	0.05496	0.04025	0.02251
Cumulative Proportion	0.4427	0.6324	0.72636	0.79239	0.84734	0.88759	0.91010

	PC8	PC9	PC10	PC11	PC12	PC13	PC14
Standard deviation	0.7811	0.7182	0.6585	0.6041	0.5511	0.5000	0.4500
Proportion of Variance	0.0192	0.0174	0.0161	0.0146	0.0134	0.0125	0.0114
Cumulative Proportion	0.9293	0.9467	0.9628	0.9774	0.9908	1.0000	1.0000

Standard deviation	0.69037	0.6457	0.59219	0.5421	0.51104	0.49128	0.39624
Proportion of Variance	0.01589	0.0139	0.01169	0.0098	0.00871	0.00805	0.00523
Cumulative Proportion	0.92598	0.9399	0.95157	0.9614	0.97007	0.97812	0.98335
	PC15	PC16	PC17	PC18	PC19	PC20	PC21
Standard deviation	0.30681	0.28260	0.24372	0.22939	0.22244	0.17652	0.1731
Proportion of Variance	0.00314	0.00266	0.00198	0.00175	0.00165	0.00104	0.0010
Cumulative Proportion	0.98649	0.98915	0.99113	0.99288	0.99453	0.99557	0.9966
	PC22	PC23	PC24	PC25	PC26	PC27	PC28
Standard deviation	0.16565	0.15602	0.1344	0.12442	0.09043	0.08307	0.03987
Proportion of Variance	0.00091	0.00081	0.0006	0.00052	0.00027	0.00023	0.00005
Cumulative Proportion	0.99749	0.99830	0.9989	0.99942	0.99969	0.99992	0.99997
	PC29	PC30					
Standard deviation	0.02736	0.01153					
Proportion of Variance	0.00002	0.00000					
Cumulative Proportion	1.00000	1.00000					

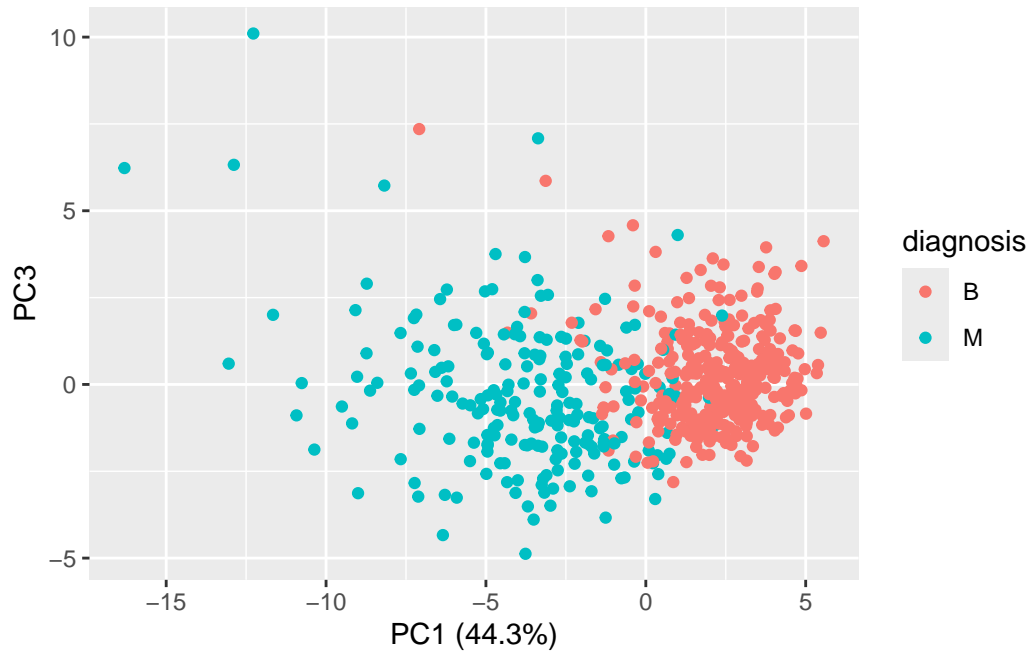
Let's make the main PC1 vs PC2

```
ggplot(wisc.pr$x) +
  aes(PC1, PC2, col=diagnosis)+
  geom_point()+
  xlab("PC1 (44.3%)" )+
  ylab("PC2 (19%)")
```



PC1 vs PC3

```
ggplot(wisc.pr$x)+  
  aes(PC1, PC3, col = diagnosis)+  
  geom_point()+  
  xlab("PC1 (44.3%)")
```



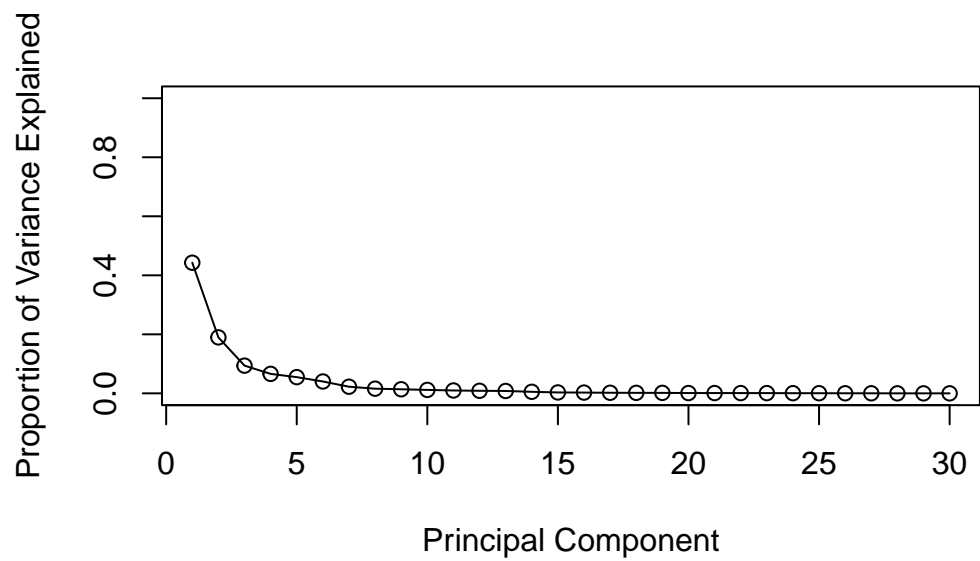
```
ylab = ("PC3 (9.3%)")
```

Q. Calculate the variance

```
pr.var <- (wisc.pr$sdev^2)  
head(pr.var)
```

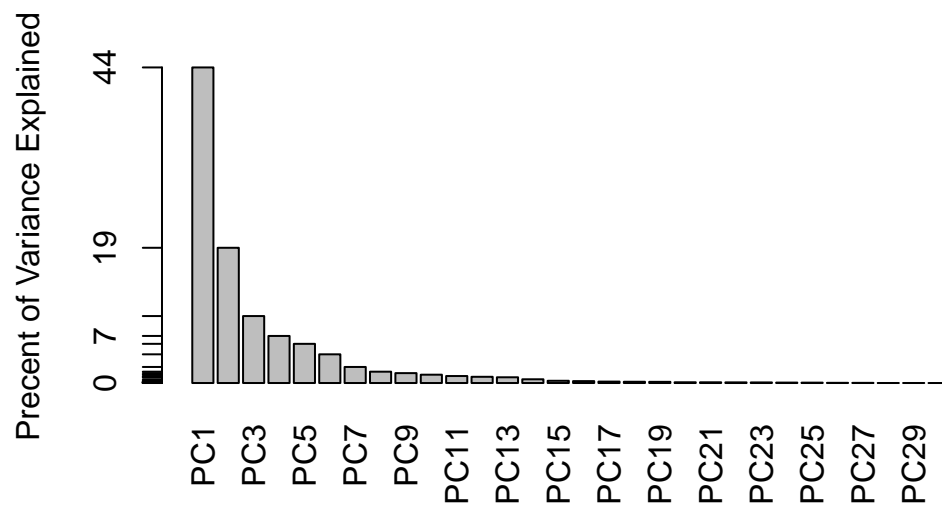
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
pve <- (pr.var) / (sum(pr.var) )  
plot(pve, xlab = "Principal Component",  
     ylab = "Proportion of Variance Explained",  
     ylim = c(0, 1), type = "o")
```



Let's make it a bar plot

```
barplot(pve, ylab = "Precent of Variance Explained",  
        names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)  
axis(2, at=pve, labels=round(pve,2)*100 )
```

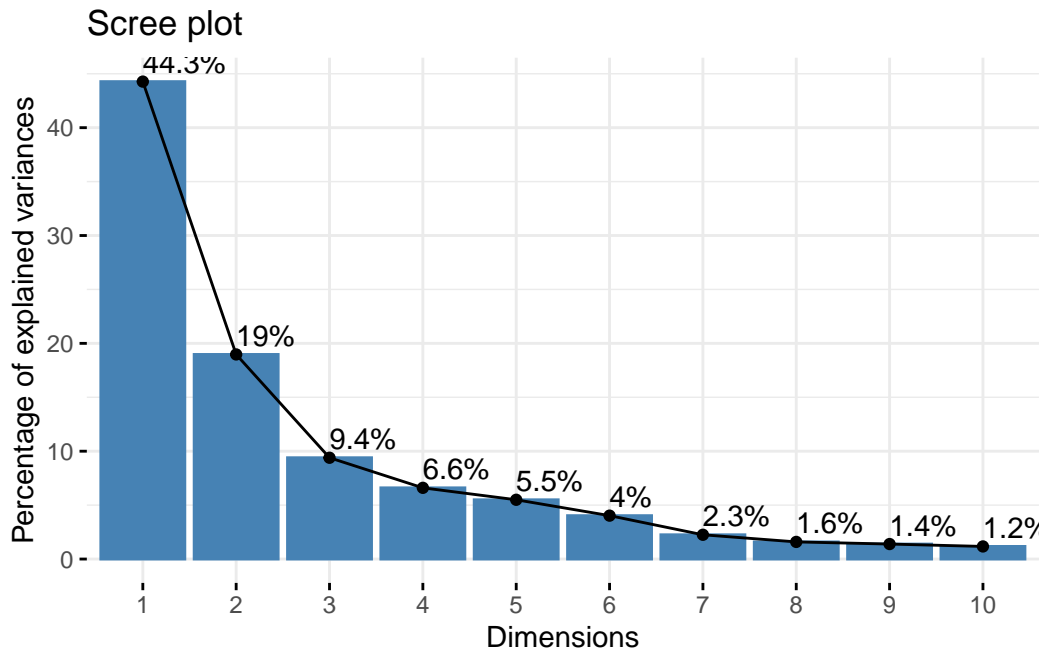


Let's make it more visually pleasing using ggplot!

```
library(factoextra)
```

Welcome! Want to learn more? See two factoextra-related books at <https://goo.gl/ve3WBa>

```
fviz_eig(wisc.pr, addlabels = TRUE)
```



Q9. For the first principal component, what is the component of the loading vector (i.e. `wisc.pr$rotation[,1]`) for the feature `concave.points_mean`?

```
wisc.pr$rotation["concave.points_mean", 1]
```

```
[1] -0.2608538
```

Q10. What is the minimum number of principal components required to explain 80% of the variance of the data?

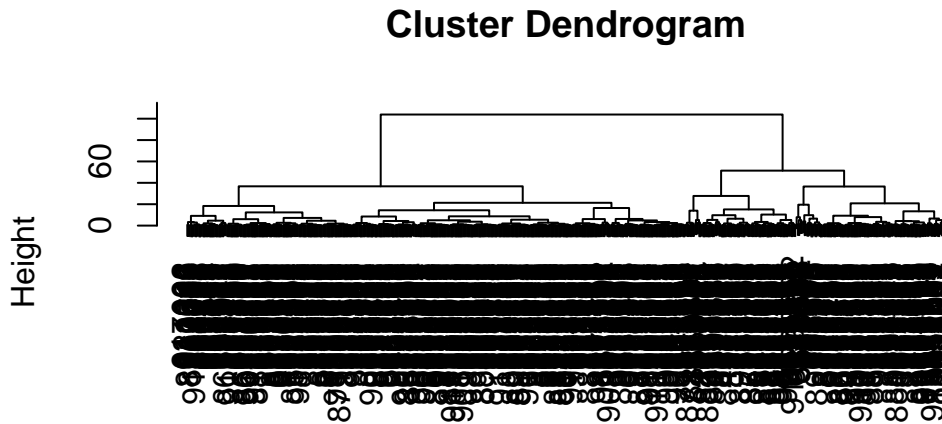
Looking at the `summary(wisc.pr)`, the minimum number of principal components required to explain 80% of the variance of data is 5, because PC5 has 84.7% of cumulative proportion.

## Combining Methods

We can take our PCA results and use them as a basis set for other analysis such as clustering.

## Clustering on PCA Results

```
wisc.pr.hclust <- hclust( dist(wisc.pr$x[, 1:2]), method = "ward.D2")  
plot(wisc.pr.hclust)
```



```
dist(wisc.pr$x[, 1:2])  
hclust (*, "ward.D2")
```

We can “cut” this tree to yield our clusters (groups):

```
pc.grps <- cutree(wisc.pr.hclust, k=2)  
table(pc.grps)
```

```
pc.grps  
 1    2  
195 374
```

How do my cluster grps compare to the expert diagnosis

```
table(diagnosis, pc.grps)
```

```
      pc.grps  
diagnosis 1    2  
B      18 339  
M     177   35
```



```
table(diagnosis)
```

```
diagnosis
  B    M
357 212
```

Q15. How well does the newly created model with four clusters separate out the two diagnoses?

There is a better separation of the B and M diagnoses, giving us a better match of the expert vs PCA data.

Q16. How well do the k-means and hierarchical clustering models you created in previous sections (i.e. before PCA) do in terms of separating the diagnoses? Again, use the `table()` function to compare the output of each model (`wisc.km$cluster` and `wisc.hclust.clusters`) with the vector containing the actual diagnoses.

They did very bad. We did much better after PCA, with the new PCA variables giving us a much better separation of M and B.

## Prediction

We can use our PCA model for the analysis of new “unseen” data. In this case from U. Mich.

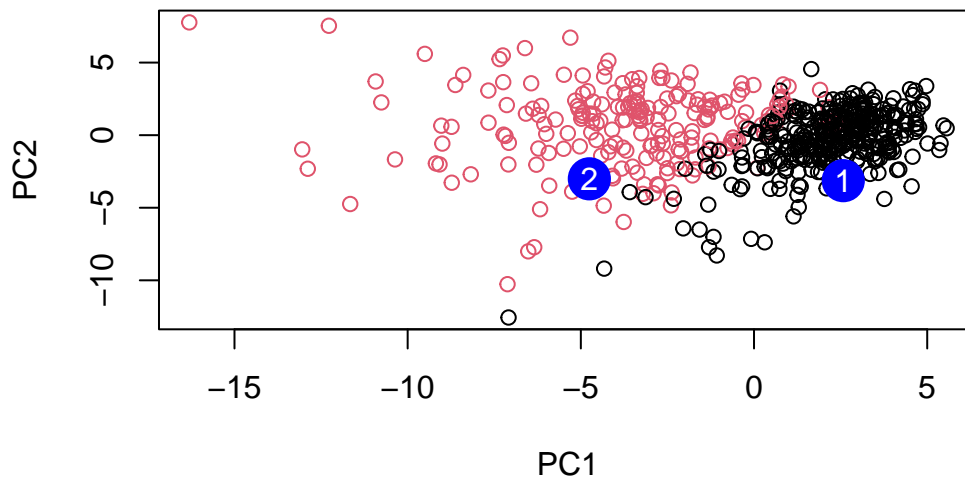
```
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr, newdata=new)
npc
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
[1,]	2.576616	-3.135913	1.3990492	-0.7631950	2.781648	-0.8150185	-0.3959098
[2,]	-4.754928	-3.009033	-0.1660946	-0.6052952	-1.140698	-1.2189945	0.8193031
	PC8	PC9	PC10	PC11	PC12	PC13	PC14
[1,]	-0.2307350	0.1029569	-0.9272861	0.3411457	0.375921	0.1610764	1.187882
[2,]	-0.3307423	0.5281896	-0.4855301	0.7173233	-1.185917	0.5893856	0.303029
	PC15	PC16	PC17	PC18	PC19	PC20	
[1,]	0.3216974	-0.1743616	-0.07875393	-0.11207028	-0.08802955	-0.2495216	
[2,]	0.1299153	0.1448061	-0.40509706	0.06565549	0.25591230	-0.4289500	
	PC21	PC22	PC23	PC24	PC25	PC26	
[1,]	0.1228233	0.09358453	0.08347651	0.1223396	0.02124121	0.078884581	
[2,]	-0.1224776	0.01732146	0.06316631	-0.2338618	-0.20755948	-0.009833238	

	PC27	PC28	PC29	PC30
[1,]	0.220199544	-0.02946023	-0.015620933	0.005269029
[2,]	-0.001134152	0.09638361	0.002795349	-0.019015820

Let's plot this!

```
plot(wisc.pr$x[,1:2], col=diagnosis)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], c(1,2), col="white")
```



Q18. Which of these new patients should we prioritize for follow up based on your results?

Based on the results, Patient 1 should be prioritized for follow-up because it has a high positive PC1 score, meaning it is more likely a malignant diagnosis.